

METHOD AND SYSTEM FOR CONTROL SYSTEM SOFTWARE

TECHNICAL FIELD OF THE INVENTION

[0001] The present invention relates to methods for displaying user control information on a touch panel and more particularly, the present invention relates to methods for building interactive control and non-interactive, non-control informational displays onto a touch panel screen.

BACKGROUND OF THE INVENTION

[0002] Touch panels are electronic devices that use liquid crystal displays (LCDs) to display functional information to a user. A touch panel is used to operate various pieces of equipment. For example, touch panels can be used to control the lighting system of a room, electronics equipment such as VCRs, DVDs, stereos, and the like, and video surveillance equipment, such as video cameras, and other similar types of equipment.

[0003] A touch panel operates by allowing the user to press a region of the LCD panel to operate an associated piece of equipment. The user knows the region of the LCD to touch because an informational display (sometimes referred to by those in the industry as “buttons”) is presented to the user that allows the user to understand the appropriate location or region at which the panel should be touched. These informational displays are usually in the form of icons for the user to press. The icons are typically icons associated with the operation of the device. For example, a VCR may have a button for the play function and a button for the recording function of the VCR. These buttons would be displayed to the touch panel user in manner that would mimic their appearance on the remote control device for the VCR.

[0004] A problem arises, however, when multiple buttons are displayed on the touch panel in close proximity to each other. When this occurs, a single touch by a user can activate more than one button. This causes either an error to occur with the touch panel, or the touch

panel itself prioritizes the buttons and selects the button with the higher priority to operate, often resulting in misoperation of the touch panel system. Therefore, it is desired to be able to build a touch panel display that allows for informational displays to be placed in close proximity to one another without resultant interference in the operations of the informational displays.

[0005] It has also been found advantageous to allow informational displays to be animated. Animation of an informational display enables certain control features to be displayed directly to the user to inform the user that the component associated with the informational display is in continuous operation. For example, an alarm system for a home may have an associated informational display that is a siren with a police light. By allowing animation, the police light can be continuously illuminated in a rotational manner to mimic a police car's lights. It is therefore desirable to have a touch panel capable of building animated informational displays.

SUMMARY OF THE INVENTION

[0006] The invention is directed to a method and system for building and displaying information on a touch panel, wherein the information may be control or non-control information, active or non-interactive information or some combination thereof. One embodiment of the invention, providing a method for building and displaying interactive, control, non-interactive and non-control information, selectively, on a touch panel comprises the steps of constructing the page for display on a touch panel, configuring a plurality of graphical display units in association with the page for display on the touch panel, and displaying the page and one or more of the graphical display unit, selectively, on the touch panel. The method further comprises the step of determining whether any of the graphical display units is activated by a touch message comprising an amount of pressure applied to a location on the touch panel.

[0007] In a preferred embodiment of the above method, each of the graphical display units is configured to have a total touch area comprising an active touch area and a non-active area, the active touch area of one graphical display unit overlapping a non-active area of another graphical display unit when both of the graphical display units are displayed. The

determining step comprises determining whether the touch message location is within the active touch area of any graphical display unit on the touch panel, and if so, activating such graphical display unit.

5 [0008] In a further embodiment, each of the graphical display units is selectively provided with one or more colors, and is further provided with one or more features selected from a group of features that includes a bitmap, an icon, text and a border. Preferably, one or more sub-pages are configured in association with the page, at least one of the sub-pages comprising a video feed. Preferably also, at least one of the graphical display units comprises a bar graph having two or more states.

10 [0009] Another embodiment of the invention comprises a computer system provided with a processor and a computer readable medium operable to build and display interactive, control, non-interactive and non-control information, selectively, on a touch panel, in accordance with the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

15 FIGURE 1 is a system schematic diagram illustrating communications between components in a network in accordance with the present invention;

FIGURE 2 is a system schematic diagram illustrating the relationships between internal components of a touch panel in accordance with the present invention;

20 FIGURE 3 is a system block diagram illustrating the information contained within a page file of the present invention;

FIGURE 4A-4D collectively provide a flow diagram of the general steps of creating and displaying a page on a touch panel in accordance with the present invention;

FIGURE 5 is a flow diagram of the general steps for determining the state of a graphical display unit of the present invention;

FIGURE 6 is a flow diagram for the general steps of determining the state of a graphical display unit comprising a bargraph in accordance with the present invention;

FIGURE 7 is a diagram showing the construction of a graphical display unit of the present invention;

5 FIGURE 8 is a flow diagram of the general steps for building the state of a graphical display unit of the present invention;

FIGURE 9 is a flow diagram of the general steps for providing a bitmap for a graphical display unit of the present invention;

10 FIGURE 10 is a flow diagram of the general steps for providing an icon for a graphical display unit of the present invention;

FIGURE 11 is a flow diagram of the general steps for providing text for a graphical display unit of the present invention;

FIGURE 12 is a flow diagram of the general steps for providing a line of text in accordance with the present invention;

15 FIGURE 13 is a flow diagram of the general steps for providing a border onto a graphical display unit of the present invention;

FIGURE 14 is a diagram showing the active touch area of a graphical display unit of the present invention;

20 FIGURE 14A is a diagram showing the active touch operation of a single graphical display unit in accordance with the present invention;

FIGURE 14B is a diagram showing the active touch operation of a touch panel having an array of graphical display units in accordance with the present invention;

FIGURE 15 is a flow diagram of the general steps of determining if a graphical display unit has been actively touched in accordance with the present invention;

FIGURE 16 is a flow diagram of the general steps of determining the type of graphical display unit actively touched in accordance with the present invention;

5 FIGURE 17 is a flow diagram of the general operation of a general graphical display unit of the present invention;

FIGURE 18 is a flow diagram of the general operation of a multi-state general graphical display unit of the present invention;

10 FIGURE 19 is a flow diagram of the general operation of a bargraph graphical display unit of the present invention;

FIGURE 20 is a flow diagram of the general operation of a multi-state bargraph graphical display unit of the present invention; and

FIGURE 21 is a flow diagram of the general operation of a joystick graphical display unit of the present invention.

15 DETAILED DESCRIPTION OF INVENTION

[0010] Figure 1 shows the interactions and communications between components of a control system 10, wherein the system 10 includes one or more computers 15 in communication with one or more master controllers 20. The master controller 20 is in communication with one or more touch panels 25. The touch panels 25 may be desktop touch panels or wall display units. The master controller 20 may be attached to various components, so that respective components may be controlled by the control system 10 through master controller 20. These components may include, but are not limited to, lighting fixtures 22a, alarm systems 22b, video equipment 22c, household electronics equipment 22d, security systems 22e, and the like.

20

[0011] A user utilizes the computer 15 to create touch panel displays such as 24 and 26 for display by the touch panel. These touch panel displays can be configured to control components 22a-e of the type referred to above. More particularly, a touch panel display comprises or contains a graphical display unit (GDU) 27, which is a symbol or other viewable indication located at a specific region upon screen 28 of touch panel 25. Control is implemented by touching the screen 28 at the specific region. Once a user creates a touch panel display, a file containing the information relating to the touch panel display is transferred from the user computer 15 to the touch panel 25 via the master controller 20. After touch panel 25 receives the file, the touch panel executes the file and builds a touch panel display.

[0012] Figure 2 is a diagram showing the operational components within the touch panel. The operational components control the functions of the touch panel display. The operational components also process the information received from the user computer 15 through the master controller 20. The operational components interact with the master controller 20 to relay user instructions from the touch panel to the devices controlled by the master controller 20. The operational components include one or more graphics engines 30. The graphics engine 30 allows the touch panel to process the information received from the computer 15. The graphics engine 30 also allows the touch panel to build panels such as 24 and 26 for display on the touch screen, pages for display on the touch screen, graphical display units (GDUs), and other similar graphical information used by the touch panel. A panel is a set of pages that is used by the touch panel to display user interface information, both interactive control information, i.e. device controls, and non-interactive, non-control information. The user creates panels by means of the user computer 15.

[0013] Panels consist of one or more pages. A page is a single informational window displayed by the touch screen. Contained within the pages are one or more GDUs and one or more sub-pages. A GDU is an informational display created to allow the user of a touch panel to control the various components communicatively connected to the master controller 20. Each GDU has two or more states. A state is the operational position of the GDU. In other words and by way of example, a GDU may have two states - on and off. In the on state, the GDU may appear to be one color, while in the off state, the GDU appears as a

different color. Each state of a GDU is a different image that is built by the graphics engine 30. Thus, if a GDU has 100 states, then the GDU has 100 associated images. Sub-pages may comprise, for example, video feeds from a camera in communication with the master controller 20 and the touch panel 25. These sub-pages are displayed to the user as windows within the page itself. Sub-pages may also have an associated state.

[0014] Communicating with the graphics engine 30 is a systems directory 35 and a panel directory 40. The systems directory 35 and the panel directory also are in communication with each other. The systems directory 35 includes a pages subdirectory 45 and a graphics subdirectory 50. The pages subdirectory 45 can be used to store information concerning the various resolution LCD screens used to build touch panels. It will be appreciated that one of ordinary skill in the art would recognize the various resolutions associated with commercially available LCD screens. The graphics subdirectory 50 contains information concerning known images, fonts, and other similar graphics information.

[0015] The panel directory 40 includes a fonts subdirectory 55, an images subdirectory 60, and a sounds subdirectory 65. The fonts subdirectory 55 contains standard fonts recognized and stored by the graphics engine 30. The images subdirectory 60 contains various images, such as bitmaps, icons, and other similar types of images recognized and stored by the graphics engine 30. The sounds subdirectory 65 contains all sounds recognized and stored by the graphics engine 30. The panel directory 40 also includes a pages file 70. The pages file 70 is created when the user uploads panel files from the user computer 15. After the upload, the pages file 70 will contain all information necessary for the graphics engine 30 to make the pages, GDUs, and sub-pages respectively available to the touch panel screen.

[0016] Figure 3 shows a schematic breakdown of the page file 70 located within the panel directory 40. The page file 70 can contain page information files concerning one or more pages, such as those depicted as 75, 80, and 85 respectively. Each page information file, 75, 80, and 85, includes instructional information to be used by the graphics engine 30 to build and display pages, GDUs, and sub-pages. The instructional information can include information concerning the state of the page as shown by block 95. Block 90 shows that the instructional information can also include information concerning the GDUs. Contained with

the GDU information is information concerning the state of the GDU, as shown by block 100. From the information contained within the page information file, a page, and the GDUs and other components of the page, can be built and displayed.

[0017] Figures 4A-4D show the method by which a touch panel builds a page. Figure 4A provides a flow diagram of the general operational steps of the touch panel. Block 200 shows that once the page files are established, the graphics engine selects a page for building, based upon user established settings and/or user commands. The graphics engine first provides the state of the page, as shown by block 202. After the graphics engine provides the state of the page, the graphics engine supplies the GDUs needed for the page selected, as shown by block 204. Once the GDUs are supplied, the graphics engine supplies the sub-pages needed for the page selected as shown by block 206. It should be appreciated that each page can contain one or more GDUs and one or more sub-pages. Upon constructing or creating the page together with its GDUs and sub-pages, the graphics engine displays the page. The GDUs and sub-pages may be displayed either simultaneously with the page or upon command of the touch panel user. Figure 4B further details the operational step of providing the state of the page as shown at block 202 in Figure 4B. Once a page is selected, the graphics engine builds the state of the page as shown by block 210. After building the state of the page it is saved to the page cache as shown by block 212. The graphics engine next sets the opacity of the page to 100% opacity, as shown by block 214. The graphics engine then stores the page data created into random access memory (RAM), as shown by block 216. Once the operational steps of providing the state of the page are completed, the graphics engine turns to supplying GDUs associated with the page, as shown by block 218

[0018] Figure 4C further details the operational step of supplying the GDUs associated with the page, as shown by block 204 of Figure 4A. Initially, the graphics engine determines the GDUs required for display with the pages as shown by block 220. The graphics engine then selects a GDU to build, as shown by block 222

[0019] After a GDU is selected, the graphics engine determines the state of the GDU and builds that state, as shown by blocks 224 and 226. Once the state of the GDU is built, the graphics engines stores the state of the GDU in a GDU cache, as shown by block 228. The

GDU cache is maintained within touch panel memory. After the state of the GDU is saved to the GDU cache, the graphics engine sets the overall opacity of the GDU, as shown by block 230. The opacity of the GDU determines the transparency of the GDU. In other words, some GDUs may be solid and some GDUs may allow objects underneath the GDU to be
5 seen.

[0020] After setting the GDU opacity, block 232 shows that the graphics engine delivers the GDU data to RAM of the touch panel. After delivering the GDU data to RAM, the graphics engine determines whether other GDUs need building, as shown by block 234. If other GDUs require building, then the steps of the process are repeated beginning with block
10 222. If no other GDUs require building, then the graphics engine proceeds to supply sub-pages associated with pages as shown by block 206 of Figure 4A. A sub-page may be, for example, a video window, or a window with operational GDUs.

[0021] In supplying sub-pages, the graphics engine first determines if the page has associated sub-pages, as in block 236 of Figure 4D. If the page has associated sub-pages, the
15 graphics engine selects a sub-page to build, as shown by block 238. Next, the graphics engine determines the state of the sub-page, as shown by block 240. After determining the state of the sub-page, the graphics engine builds the state of the sub-page, as shown by block 242. The state of the sub-page is then saved to the sub-page cache, as in block 244. Once the sub-page is saved, the graphics engine selects the opacity of the sub-page, as in block
20 246. Thereafter, the need for associated GDUs is considered, block 248. If another GDU is needed, the graphics engine then determines the GDUs needed for the sub-pages as in block 250. The graphics engine then selects a sub-page GDU to build, as in block 252. Once selected, the graphics engine determines the state of the sub-page GDU, as in block 254, and the state of the sub-page GDU is built, as in block 256. After building the sub-page GDU,
25 the sub-page GDU is saved to the GDU cache as in block 260. Next, the opacity of the GDU is selected, as in block 262. Once the graphics engine selects the opacity, the graphics engine stores the sub-page GDU data in RAM, as in block 264. The graphics engine then determines whether other sub-pages exist which require building, as shown by block 266. If other sub-page GDUs exist, then the steps of the process for building sub-page GDUs are
30 repeated beginning with block 252. If no other sub-page GDUs require building, then the

graphics engine stores the sub-page data to RAM, as shown by block 268. Next, the graphics engine determines if other sub-pages need building, as shown by block 270. If other sub-pages require building, then the graphics engine repeats the steps described above beginning with block 238.

5 **[0022]** Figures 5-6 provide a more thorough description of block 224 of Figure 4 for determining the initial state of a GDU. Figure 5 shows the general steps in determining the state of a GDU. To begin, the type of GDU is determined. Although Figure 5 shows six specific types of GDUs, it will be appreciated by one of skill in the art that other types of GDUs may exist. In determining the type of GDU to be built, each type of GDU is
10 examined, if necessary. Under this process, the graphics engine determines whether the GDU is a general GDU, as shown by block 500. If the GDU is a general GDU, that is, has only two possible states, then the graphics engine builds the state of the general GDU as shown by block 505. If the GDU is not a general GDU, then the graphics engine determines if the GDU is a multi-state general GDU, that is, has more than two states, as in block 510. If
15 the GDU is a multi-state general GDU, then the graphics engine builds the state of the multi-state general GDU as shown by block 515. If the GDU is not a multi-state general GDU, that is, may have more than two states, then the graphics engine determines whether the GDU is a bargraph GDU, as shown by block 520. If the GDU is a bargraph GDU, then the graphics engine builds the GDU based upon the steps shown in Figure 6, as shown by block 525. If
20 the GDU is not a bargraph GDU, the graphics engine then determines whether the GDU is a multi-state bargraph GDU, as shown by block 530. If the GDU is a multi-state bargraph GDU, then the graphics engine builds the multi-state bargraph GDU built as shown by block 535. If the GDU is not a multi-state bargraph GDU, the graphics engine then determines whether the GDU is a joystick GDU, as shown in block 540. If the GDU is a joystick GDU,
25 the graphics engine builds the joystick GDU, as shown by block 545. If the GDU is not a joystick GDU, then an error message is displayed, as shown by block 560. It should be appreciated that other GDUs may exist that would continue the process until either the correct GDU is found or an error message is displayed.

30 **[0023]** Figure 6 further illustrates the process for determining the state of a bargraph general GDU. Once a bargraph GDU is detected, the graphics engine determines the two

states of the bargraph GDU, as in block 602. After the states are determined, block 605 shows that the graphics engine determines the display level for the bargraph GDU. A bargraph GDU has two associated states. The display level of a bargraph GDU can be represented as a composite of each of the two associated states. For example, a horizontally oriented bargraph (not shown) could have a first state made up of 100% blue, and a second state made up of 100% red. The first and second states could be combined so that the bargraph is a composite of the first state and the second state wherein, for example, 75% of its area is blue and 25% of its area is red.

[0024] After the display level is determined, block 610 shows that the graphics engine generates a composite that is a percentage of each state associated with the bargraph GDU. Next, the graphics engine determines if the bargraph GDU has a slider, as shown by block 615. A slider is a graphical representation used to depict the display level position of the bargraph GDU. If a slider exists, a representation thereof is generated, as shown in block 617. Once blocks 605 through 615 are completed, the bargraph GDU is built, as shown by block 620. For the above example, the slider would be shown at a horizontal position along the bargraph so that 75% of the area of the bargraph, on one side of the slider, would be blue, and 25% of the area, on the other side, would be red.

[0025] Figures 7-13 further explain block 226 of Figure 4C for building the state of a GDU. Figure 7 shows a diagram illustrating the construction of a GDU. As shown in Figure 7, the GDU is first sized to the appropriate dimensions, as shown by 700. After sizing the graphical display unit and adding color, a bitmap 705 is generated for placement within the GDU dimensions 700. The actual construction and placement of the bitmap 705 will be discussed hereinafter. An icon 710 can next be generated for placement within the GDU 700. Again, the actual capturing and placement of the icon will be discussed hereinafter. After the icon is generated, lines of text 715 are positioned on the GDU 700. Finally, a border 720 can be generated for the GDU 700. The construction and placement of the lines of text 715 and the border 720 are discussed infra. The result of these steps is a fully constructed GDU 725. It will be appreciated that not every GDU will include a bitmap, icon, text and a border. It will also be appreciated that while this embodiment of the invention uses these components, other components may exist that could be provided for the GDU.

[0026] Figure 8 shows the general process for building the state of a GDU. Once the type and state of the GDU are determined, the graphics engine analyzes the GDU cache to determine if the GDU has been previously built, as shown by block 800. The GDU cache is a database that stores all GDU states previously built. If the GDU has been previously built, then it is recalled from the GDU cache, as shown by block 805. If the GDU is not in the GDU cache, then the graphics engine creates a transparent image of the GDU, as shown by block 810. The transparent image provides the relative dimensions of the GDU. After the transparent image is created, block 815 shows that the graphics engine determines the color, or colors associated with the GDU and fills the transparent image with the appropriate colors.

[0027] Next, the graphics engine determines whether the GDU has an associated bitmap, as shown by block 820. If the GDU has an associated bitmap, the bitmap is retrieved from a bitmap database stored in RAM and furnished to the GDU, as shown by block 825. Once the graphics engine begins operating the page file in the panel directory, the graphics engine creates the bitmap database. If the GDU does not contain a bitmap or once the bitmap has been retrieved and furnished, then the graphics engine determines whether the GDU has an associated icon, as shown by block 830. If the GDU contains an icon, then the icon is retrieved and furnished to the GDU, as shown by block 835. If the GDU does not contain an icon or once the icon has been retrieved and furnished, then the graphics engine determines whether the GDU has any associated text, as shown by block 840. If the GDU has an associated line of text, then the line of text is built and furnished to the GDU, as shown by block 845. If the GDU does have text or once the text has been built and furnished, then the graphics engine determines whether the GDU has an associated border, as shown by block 850. If the GDU has a border, then the border is built and furnished to the GDU, as shown by block 855. If the GDU does have a border or once the border has been built and furnished, then the built GDU is saved to the GDU cache, as shown by block 860.

[0028] Figures 9–13 further illustrate processes for creating a GDU through the creation of bitmaps, icon, text, and borders. Once it is determined that a bitmap is needed, the dynamic nature of the bitmap is determined, as shown by block 900, in Figure 9. A dynamic bitmap is one that is retrieved from a remote location, for example the Internet, a remote storage

device, or other similar means, rather than being stored in the images subdirectory 60 of the panel directory 40 of Figure 2.

[0029] If it is determined that the bitmap is dynamic, then the graphics engine retrieves the bitmap from the remote location where it is stored using the master controller 20 as shown in
5 block 905. Once the bitmap is retrieved from the remote location, the graphics engine determines the location of the bitmap on the GDU, as shown by block 915. After the location of the bitmap is determined, the bitmap is provided or delivered to the GDU, as shown by block 920. If the bitmap is not dynamic, then block 910 shows that the graphics engine retrieves the bitmap from the images subdirectory 60. Once the bitmap is retrieved
10 from the images subdirectory, then block 915 shows that the graphics engine determines the location of the bitmap on the GDU. After determining the location of the bitmap, the bitmap is delivered to the GDU, as shown by block 920.

[0030] Once it is determined that an icon is associated with the GDU, the graphics engine retrieves the icon from an icon database contained in RAM, as shown by block 1000 of
15 Figure 10. The icon database is created once a page file becomes operational. The icon database is indexed such that each icon needed for a GDU can be found by referring to its place in the index. Once the icon is retrieved from the database, block 1005 shows that the graphics engine determines the location of the icon on the GDU. After determining the location of the icon, the icon is delivered to the GDU, as shown by block 1010.

20 [0031] Once a determination is made that the GDU contains a line of text, block 1100 of Figure 11 shows that the graphics engine determines the font associated with the line of text. Once the appropriate font is determined, the graphics engine determines whether the font exists in the font database, as shown by block 1105. Block 1110 shows that if the font is not found, then a default font is chosen and the default font is retrieved from the font database, as
25 shown by block 1115. If the font does exist in the font database, then the graphics engine retrieves the font from the font database, as shown by block 1115. Once the font is retrieved, the line of text is sized, as shown by block 1120. Determining the size of the text assists in determining the location of the text on the GDU. After determining the size of the text, the graphics engine determines the location of the text on the GDU, as shown by block 1125.

After determining the location of the text on the GDU, the line of text is created and positioned on the GDU, as shown by block 1130. Once a line of text is positioned on the GDU, the graphics engine determines whether another line of text is associated with the GDU, as shown by block 1135. If another line of text is associated with the GDU, then
5 blocks 1100 through 1130 are repeated. These steps are repeated until all lines of text have been created and positioned onto the GDU, as shown by block 1140.

[0032] Figure 12 further illustrates the provision or creation of a line of text, as shown by block 1130 in Figure 11. In providing a line of text, the graphics engine determines whether the text has an associated text effect. A text effect is a special feature of the text that makes
10 the text more prominent or changes the appearance of the text, i.e., gives the text depth. An example of a text effect would be a shadowing feature. If the line of text has an associated effect, then the graphics engine determines the first character in the line of text, as shown by block 1205. Once the character is determined, the graphics engine determines whether the character is an end of line character, as shown by block 1210. An end of line character is a
15 marker that specifies the completion of a line of text.

[0033] If the character is not an end of line character, then the graphics engine determines whether the character with the appropriate effect is stored in the effect cache, as shown in block 1220. The effect cache is a database containing all characters with effects that have been previously built. If the character with the appropriate effect is stored in the effect cache,
20 then the character and effect are delivered to the GDU, as shown by block 1225 and the next character is determined, as shown by block 1205.

[0034] If the character with appropriate effect is not stored in the effect cache, then the graphics engine creates a description of the character and effect, as shown by block 1230. Block 1235 then shows that the description of the character and the effect are saved to the
25 effect cache after which the character effect is delivered to the GDU, as shown by block 1225. The graphics engine then determines the next character, as shown by block 1205.

[0035] If the character is an end of line character, then the effects for the line of text have all been provided, and the line of text is reset to the first character to generate actual text characters, as shown by block 1240. Block 1245 shows that the graphics engine again

determines the character. After the character is determined, the graphics engine again determines whether the character is an end of line character, as shown by block 1250. If the character is an end of line character, the line of text is complete, as shown by block 1255. If the character is not an end of line character, then the character cache is checked to determine if the character exists, as shown in block 1260. Like the effect cache, the character cache is created once the graphics engine has built characters. If the character is not stored in the character cache, then block 1265 shows that the graphics engine creates a character description for the character after which the character is stored in the character cache, as shown in block 1270. After the character is stored in the character cache, the character is retrieved and delivered to the GDU, as shown by block 1275 and the next character in the line of text is determined, as shown by block 1245. If the character is stored in the character cache, then the character is retrieved and delivered to the GDU, as shown by block 1275. The next character in the line of text is determined, as shown by block 1245.

[0036] Once it is determined that the GDU has an associated border, the border description is retrieved from the border database in RAM, as shown by block 1300 of Figure 13. After the border description is retrieved, the graphics engine masks the GDU, as shown by block 1305. Masking of the GDU allows for the GDU to be shaped appropriately. All GDUs begin as squares. It is through the masking, or cropping process, that a GDU can be formed into other shapes.

[0037] After the GDU has been masked, the graphics engine determines whether the border is stored in the border cache, as shown by block 1310. The border cache is created once the graphics engine begins building the borders for GDUs. If the border is stored in the border cache, then the border is retrieved and delivered to the GDU, as shown by block 1315. If the border is not stored in the border cache, then the border pieces are created, as shown by block 1320. Each border is a series of eight or more border pieces. Each piece will cover a particular region of the GDU. Once the border pieces are created block 1325 shows that the border pieces are colorized. Next, the colorized border pieces are stored to the border cache, as shown by block 1330. After storing the border pieces in the border cache, the border pieces are delivered to the GDU, as shown by block 1315.

Operation of Touch Panel

[0038] In the operation of touch panels, it has become important to be able to furnish GDUs to the touch panel, such that the GDUs emulate the operational controls for a device associated with and represented by the GDU. For example, a Digital Video Device (DVD) may have a remote control that includes a play button, a reverse button, a fast forward button and a pause button, all within the same general region. A touch panel, to achieve user-friendly operation, should emulate the configuration of the DVD remote control device.

[0039] Because all GDUs begin as a square configuration and are then masked or cropped to their display shape, GDUs within close proximity may cause interference with the operations of one another. More particularly, a touch message is the application of pressure to a region of a touch panel screen on which a GDU is displayed. The application of pressure generates a control signal for a purpose associated with the GDU. However, because a GDU remains square even after masking, the areas of adjacent GDUs may overlap, if the GDUs are in close proximity. Thus, a touch message intended for a particular GDU could also apply pressure to a closely spaced adjacent GDU. Accordingly, to prevent this interference, it has been found that an active touch area can be defined for each GDU, for use in discerning between GDUs in close proximity. The active touch area of a GDU will generally be the unmasked area of its corresponding square.

[0040] Figure 14A shows a diamond shaped GDU 1400 with an active touch area 1405. The diamond shaped GDU 1400 is considered square in shape by the touch panel and has a total touch area shown by the dotted square 1410. However, the active touch area 1405 is defined to be only the area within the perimeter of the GDU 1400, with the perimeter being defined by lines 1415, 1420, 1425, and 1430, respectively.

[0041] Once a touch message is received by the touch panel, the touch panel determines if the touch message actively touches a GDU. Thus, in Figure 14A a touch message 1435 is received by the touch panel in the proximity of the GDU 1400, and more specifically within its total touch area 1410. The touch panel mathematically orients a hypothetical line, depicted as YX, between the center (Y) of the GDU and the location of the touch message (X). Next, the touch panel, using the line YX as orientation, mathematically generates a line

from the touch message (X) to the edge (A) of the GDU. The touch panel then analyzes each point along the line XA, beginning at the touch message point (X), until the point at the edge of the square (A) is reached. If any point along XA corresponds to the active touch area 1405, then the touch panel activates or operates that GDU. On the other hand, if a point
5 along XA is within total touch area 1410 but not active touch area 1405, the touch panel does not operate that GDU. Figure 14A shows the message 1435 to be in such non-active area of the GDU 1400. This process allows for GDUs to be positioned on a page such that the minimum spacing between the GDUs may be no more than one pixel in resolution. This process also allows for GDUs to be positioned in close proximity to each other without the
10 detrimental effect of interference due to overlapping areas. Such a configuration is shown in Figure 14B.

[0042] Figure 14B shows four GDUs of the configuration illustrated in Figure 14A. Four GDUs, labeled (I), (II), (III), and (IV), are arranged such that the total touch areas of the GDUs overlap. More importantly, the GDUs are arranged in a manner such that the total
15 touch area of one GDU overlaps with the active touch area of another. For example, GDU (I) has an active touch area 1440 and a total touch area 1448. GDU (II) has an active touch area 1442 and a total touch area 1450. As can be seen by this configuration, the total touch area 1448 of GDU (I) overlaps with the active touch area 1442 of GDU (II) at location 1456. Such an overlap can be seen again with reference to GDU (III) and GDU (IV). GDU (III)
20 has an active touch area 1444 and a total touch area 1452. GDU (IV) has an active touch area 1446 and a total touch area 1454. Again, here the total touch area 1454 of GDU (IV) overlaps with the active touch area 1444 of GDU (III) at location 1458. Thus, as can be seen, certain configurations of GDUs will provide overlapping areas of active touch and non-active touch from two separate GDUs. Therefore, a method for discerning which button to operate
25 should a touch message be received from an area of overlap is needed. The present method provides such a method. As shown in Figure 14B, a touch message 1460, shown by an X, is received by the touch panel. The touch message 1460 is located within a region of overlap where the total touch area of GDU (I) and the active touch area of GDU (IV) overlap. To discern which GDU, if any, is activated by the touch message 1460, the touch panel
30 mathematically plots a line between and the touch message location 1460 and the edge of

each GDU, as shown by Figure 14A. In Figure 14B, the lines are represented as line XB, which is the line between the touch message 1460 and the edge B of GDU (I), and as line XC, which is the line between the touch message and the edge C of GDU (IV). It should be noted that the touch message location is analyzed with respect to each GDU. However, if the touch message is outside the total touch area for a GDU, then the touch panel determines that the particular GDU has not been hit. In the present example, GDU (II) and GDU (III) fall within this category.

[0043] Once the touch panel has mathematically plotted the lines XB and XC, each line is analyzed in the manner described in Figure 14A to determine whether the touch message is in an active touch area for the respective GDUs. The order in which the lines are analyzed is the order in which the GDUs were created by the graphics engine. Thus, in this instance and by way of example only, GDU (I) and line XB would be analyzed first and GDU (IV) and line XC would be analyzed next. Upon such analysis, the touch panel would determine that the touch message 1460 actively touches GDU (IV), but does not actively touch GDU (I). Thus, GDU (IV) would be activated and the user would not experience interference problems between GDU (I) and GDU (IV).

[0044] Figures 15-21 illustrate the operation of the touch panel when a touch message is received, wherein Figure 15 generally illustrates the process undertaken by the touch panel when such event occurs. A touch message is received when a user applies pressure to a region of a touch panel display upon the touch panel screen. Once a touch message is received to select a particular page of the touch panel display, the touch panel determines and orders the sub-pages associated with the page, as shown by block 1510. If sub-pages exist, the touch panel determines if any of the sub-pages have been hit, as in block 1515. A sub-page is "hit" if a touch message is received anywhere in the total touch area of its associated GDU. If a sub-page is hit, the touch panel determines if the sub-page has associated sub-page GDUs, as in block 1520. If no sub-page GDUs exist, then the touch panel operates the sub-page as in block 1550. If sub-page GDUs exist, the touch panel determines and orders the sub-page GDUs, as in block 1525. The order of the GDUs relates to the order of creation. Next, the touch panel determines if a sub-page GDU has been hit, as in blocks 1530 and 1535. If the sub-page GDU has been actively touched, then the touch panel operates the

sub-page GDU, as in block 1540. If a sub-page GDU was the last sub-page GDU, as in block 1545, then the touch panel operates the sub-page, as in block 1550. If other sub-page GDUs exist, then the steps described above are repeated beginning at block 1525.

[0045] If the touch determines the sub-page was not hit, then the touch panel determines if that was the last sub-page, as in block 1555. If other sub-pages exist, then the touch panel begins over the process described above beginning with block 1510. If no other sub-pages exist, then it is determined if the page has GDUs as in block 1560. If GDUs exist, then all GDUs active on the page are ordered and determined if hit, as shown by blocks 1565 and 1570. If it is determined that a GDU is hit, the touch panel then determines if the GDU was actively touched, as shown by block 1575. If the GDU was actively touched, then the GDU is operated, as shown by block 1580. If the GDU was not actively touched or the GDU was not hit, the touch panel then determines whether the GDU was the last GDU on the page, as shown by block 1585. If the GDU was not the last GDU on the page, then the next GDU in the order is determined, as shown by block 1565. If the GDU was the last GDU on the page, the touch panel then determines if the touch operates the page, as shown by block 1590. A page may have operational features separate and apart from the GDUs.

[0046] Figure 16 illustrates generally the process undertaken once a GDU is actively touched. Block 1600 shows that the touch panel initially determines the type of GDU actively touched. To perform this function, the touch panel determines if the GDU was a general GDU, as shown by block 1605. If the GDU is a general GDU, then the general GDU is operated, as shown by block 1610. If the touch panel determines that the GDU is not a general GDU, then the touch panel determines whether the GDU actively touched was a multi-state general GDU, as shown by block 1615. If the touch panel determines that the GDU was a multi-state general GDU, then the multi-state general GDU is operated, as shown by block 1620. If the touch panel determines that the GDU is not a multi-state general GDU, then the touch panel determines whether the GDU actively touched was a bargraph general GDU, as shown by block 1625. If the touch panel determines that the GDU was a bargraph GDU then the bargraph GDU is operated, as shown by block 1630. If the touch panel determines that it is not a bargraph GDU, then the touch panel determines whether the GDU actively touched was a multi-state bar graph GDU, as shown by block 1635. If the touch

panel determines that the GDU was a multi-state bargraph GDU, then the multi-state bargraph GDU is operated, as shown by block 1640. If the touch panel determines that the GDU is not a multi-state bargraph GDU, then the touch panel determines whether the GDU actively touched was a joystick GDU, as shown by block 1645. If the touch panel determines that the GDU was a joystick GDU, then the joystick GDU is operated, as shown by block 1650. If the touch panel determines that the GDU is not a joystick GDU, then the touch panel checks other GDUs, or operates the page or subpage, as shown by block 1665.

[0047] Figure 17 shows the operation of an actively touched general GDU, as shown by block 1700. Once it is determined that a general GDU has been actively touched, the requested state of the general GDU is determined, as shown by block 1705. After the requested state is determined, the requested state is compared with the current state of the general GDU to see if the states are the same, as shown by block 1710. If the requested state equals the current state, then the process ends, as shown by block 1715. If the requested state does not equal the current state, then the requested state is displayed, as shown by block 1720. It should be appreciated that if the requested state has not yet been built, then the process described in the preceding figures for building the state of a GDU would be followed.

[0048] Figure 18 shows the operation of an actively touched multi-state general GDU, as shown by block 1800. Once a multi-state general GDU has been actively touched, a determination is made as to whether the multi-state general GDU is a repeating GDU, as shown by block 1805. If the multi-state general GDU is not a repeating GDU, then it is determined whether the requested state equals min, as shown by block 1810. If the requested state equals min, then it is determined whether the current state equals min, as shown in block 1815. If the current state equals min, then the operation ends, as shown by block 1820. If the current level does not equal min, then the multi-state general would change display from its current state to min, sequentially displaying each state between its current state and its min state, as shown by block 1825. It should be appreciated that if any state between the current state and min does not exist, then that state would be built in accordance with the processes and figures described above.

[0049] If the requested state does not equal min, then it is determined whether the current state equals max, as shown by block 1830. If the current state equals max, then the operation ends, as shown by block 1820. If the current state does not equal max, then the multi-state general would change display from its current state to max, sequentially displaying each state
5 between its current state and its max state, as shown by block 1835. It should be appreciated that if any state between the current state and max does not exist, then that state would be built in accordance with the processes and figures described above.

[0050] If it is determined that the multi-state general GDU is a repeating GDU, then it is determined whether the requested state equals min, as shown by block 1840. If the requested
10 state does not equal min, it is determined whether the current state equals max, as shown by block 1845. If the current state equals max, then the state of the multi-state general GDU is set to min and the multi-state general GDU continues to change display from its min state to its max state, sequentially displaying all states in-between, as shown by block 1850. It should be appreciated that this operation allows for the multi-state general GDU to be
15 continuously animated. For example, the multi-state general GDU may be built to resemble a stopwatch such that the min state equals 0 and the max state equals 60. Each state in-between the min state and the max state would account for a one second time interval. Thus, the second hand on the stopwatch would be continuously moving from 0 to 60 under the process described above.

[0051] If the current state does not equal max, then the multi-state general would change display from its current state to max, sequentially displaying each state between its current state and its max state, as shown by block 1855. Once the max state has been achieved, then the state is set to min and the multi-state general GDU continues to change display from its min state to its max state, sequentially displaying all states in-between, as shown by block
25 1850. It should be appreciated that if any state necessary for the operations described in reference to blocks 1850 or 1855 does not exist in the GDU cache, then that state would be built in accordance with the processes and figures described above.

[0052] If the requested state equals min, then it is determined whether the current state equals min, as shown by block 1860. If the current state equals min, then the operation of the

multi-state general GDU ends, as shown by block 1865. If the current state does not equal min, then the multi-state general GDU would change display from its current state to min, sequentially displaying each state between its current state and its min state, as shown by block 1870. Again, it should be appreciated that if any state between the current state and min does not exist, then that state would be built in accordance with the processes and figures described above.

[0053] Figure 19 illustrates the operation of an actively touched bargraph GDU, as shown by block 1900. Once it is determined that a bargraph GDU has been actively touched, block 1905 shows that the requested display level for the bargraph GDU is determined. A determination is then made as to whether the actual level of the bargraph GDU is equal to the requested display level, as shown by block 1910. If the actual level equals the display level, then the operation of the bargraph GDU ends, as shown by block 1915. If the actual level does not equal the display level, then the bargraph GDU is adjusted to show a percentage of its states equal to the requested display level, as shown by block 1920. For example, if the bargraph GDU represents a volume control bar and the two states of the bargraph GDU are blue and yellow, then when the requested display level equals 70%, the yellow state of the bar graph GDU would fill 70% of the available area of the bargraph GDU and the blue state would fill 30% of the available area of the bargraph GDU.

[0054] Figure 20 shows the operation of an actively touched multi-state bargraph GDU, as shown by block 2000. After determining that a multi-state bargraph GDU has been actively touched, the requested display level is determined, as shown by block 2005. Block 2010 shows that the actual level of the multi-state bargraph GDU is then compared with the requested display level. If the actual level equals the display level, then the operation of the multi-state bar graph GDU ends, as shown by block 2015. If the actual level does not equal the requested display level, then the state of the multi-state bargraph GDU representing the requested display level is displayed, as shown by block 2020. It should be appreciated that if the state representing the requested display level does not exist, then that state will be built in accordance with the processes and figures described above.

[0055] Figure 21 shows the operation of an actively touched joystick GDU, as shown by block 2100. Initially, the requested cursor X and Y values are determined, as shown by block 2105. Block 2110 shows that the current cursor X and Y values are then compared with the requested X and Y values to determine if they are equal. If the current cursor X and Y values equal the requested cursor X and Y values, then the operation of the joystick GDU ends, as shown by block 2115. If the current cursor X and Y values are not equal to the requested cursor X and Y values, then the cursor associated with the joystick GDU is moved to the requested cursor X and Y values, as shown by block 2120. It should be appreciated that to move the cursor to the requested cursor X and Y values, a different state of the joystick GDU is displayed. It should further be appreciated that if the state of the joystick GDU with the requested cursor X and Y values does not exist, then that state will be built in accordance with the processes and figures described above.

[0056] Obviously, many other modifications and variations of the present invention are possible in light of the above teachings. The specific embodiments discussed here are merely illustrative and are not meant to limit the scope of the present invention in any manner. It should be appreciated that the present invention provides many applicable inventive concepts that may be embodied in a wide variety of specific contexts. It is therefore to be understood that within the scope of the disclosed concept, the invention may be practiced otherwise than as specifically described.